



Simultaneous Localization and Segmentation of Fish Objects Using Multi-task CNN and Dense CRF

Alfonso B. Labao and Prospero C. Naval Jr. (✉)

Computer Vision and Machine Intelligence Group, Department of Computer Science,
College of Engineering, University of the Philippines, Diliman,
Quezon City, Philippines
pcnaval@dcs.upd.edu.ph

Abstract. We propose a deep learning tool to localize fish objects in benthic underwater videos on a frame by frame basis. The deep network predicts fish object spatial coordinates and simultaneously segments the corresponding pixels of each fish object. The network follows a state of the art inception resnet v2 architecture that automatically generates informative features for object localization and mask segmentation tasks. Predicted masks are passed to dense Conditional Random Field (CRF) post-processing for contour and shape refinement. Unlike prior methods that rely on motion information to segment fish objects, our proposed method only requires RGB video frames to predict both box coordinates and object pixel masks. Independence from motion information makes our proposed model more robust to camera movements or jitters, and makes it more applicable to process underwater videos taken from unmanned water vehicles. We test the model in actual benthic underwater video frames taken from ten different sites. The proposed tool can segment fish objects despite wide camera movements, blurred underwater resolutions, and is robust to a wide variety of environments and fish species shapes.

Keyword: Fish object localization

1 Introduction

1.1 Overview and Motivation

The research of this paper is motivated by the relatively old problem of fish-counting. The problem of how to accurately count estimate fish populations is important for scientists and fishermen as it provides the basis for local and national government decisions as to how much fish can be harvested from the ocean. [23]. From [5], measuring the distribution and abundance of fish organisms in marine environments is of fundamental importance in assessing an ecosystem's health, since it provides criteria for adjusting fishing quotas. If fish abundance

counts are left unchecked, it may lead to overfishing which disrupts the natural dynamics of fish populations.

However, the fish-counting problem involves the sub-routine of Fish object localization, and localization of fish objects in underwater video frames is non-trivial due to several challenges unique to underwater videos. [17] mentions that underwater mediums are characterized by light absorption and scattering caused by dissolved constituents and suspended particulates that produce marine snow artifacts. This causes several false positives on fish objects. Other problems include strong similarities between fish-object colors and background colors - which are both dominant in the blue channel. In some videos - particularly those that are taken near the surface - strong illumination changes are prevalent, which could distort methods that rely on brightness and motion for fish localization.

1.2 Related Methods

Earlier methods for fish species localization [21], rely on background-subtraction methods. This approach usually gathers background information per pixel in the form of average values in the RGB channels. Pixels that show significant deviations from the average RGB values are classified as moving objects - which are usually fish objects. A limitation of this approach is its dependence on fixed-camera set-ups and static backgrounds. But for videos with dynamically changing backgrounds - as in cases of sudden illumination caused by turbulent surface waves, or by moving cameras, background subtraction is no longer useful. Other methods rely on manually-defined features and thresholding to detect fish objects such as [11], which relied on CLAHE-thresholded features. These methods however are limited by the information set by the manually-crafted features and may not generalize to a wide variety of fish appearances and shapes. In addition, such methods do not perform in a unified manner since the feature extraction, object localization and classification methods are separate, and a single frame has to pass through different methods to predict the desired outputs.

Recent methods based on deep-learning show some promise to address the limitations of the above approaches. Particularly, deep learning combines both the feature generation process and the localization process into a unified network that could be elegantly trained via backpropagation. Since features are learned from the data, it overrides the limitations caused by manually-defined features. In addition, several features of deep learning are robust to changes in brightness or motion since they are edge-based. This overcomes the need to rely on motion-sensitive background subtractors - which would be applicable to underwater video applications.

Some prior works that use deep learning methods for fish detection is by [14] and [13]. However, both methods perform species classification with bounding-box estimation, while our method performs pixel-wise instance segmentation with bounding-box estimation. In addition, the previous works rely on manually-labeled ground-truth from the FishCLEF database [8]. The work of [12] uses a

deep learning network (Residual Network) to compute for the semantic segmentation of fish object pixels based on weak labels provided by Gaussian background-subtraction. However, the deep network model in [12] does not perform a bounding-box localization task but limits itself to semantic segmentation. A box localization sub-routine can be performed on the output of [12]’s network but it is an external sub-routine that is not embedded on the deep network and may lead to additional processing delays.

For this paper, the deep learning methods that are proposed are networks that could perform both localization, classification and segmentation tasks, [18], [3] - since the problem is to both detect and segment a large number of fishes in each video frame, and differentiate each fish instance. Hence, the proposed method can perform multiple localization and segmentation tasks all at the same time - without need to call methods external to the core architecture. This simplifies the process, and given modern GPU hardware, prediction time is lessened - enabling fast video processing of thousands of frames. To further refine predicted masks, we pass the network’s predicted mask to a dense Conditional Random Field (CRF) [9] for post-processing as is done in several other methods. Relating our model to the motivation provided in the previous sub-section, the localization sub-task provides fish-count estimates. The segmentation sub-task on the other hand provides an easier visual assessment of the fish contours which could be used for fish size estimation in future works. For our experiment, we use actual underwater video data that is subject to several underwater illumination distortions and with large numbers of fish objects per frame. We also rely on weak-labels to train our algorithm since manufacturing manual ground truth for our large data is impractical. Models trained with only weak-labels have to be more powerful to generalize well despite inherent errors in training.

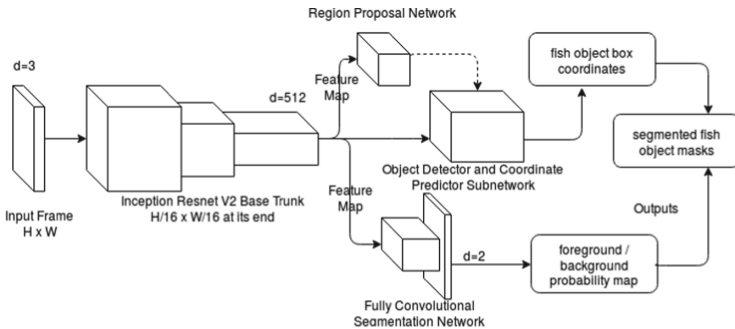


Fig. 1. Network architecture

2 Methodology

The base model of the proposed tool is a convolutional neural network (CNN). Convolutional neural networks first gained popularity in [10], which won the

ImageNet classification task [19] by accurately classifying 1000 classes with 90% accuracy. Recent models extended the classification CNN to localization tasks, where several objects can be detected per input frame, and the CNN localizer outputs cartesian box coordinates (x_1, y_1, x_2, y_2) per object. CNNs are trained through the backpropagation algorithm and several iterations are required before achieving meaningful results. For this paper, the CNN follows an architecture [3] that can simultaneously perform several tasks. The only required input during is an RGB video-frame during test-time (of arbitrary size), and the output is a list of fish-objects, along with their corresponding box coordinates, and instance masks (where an instance mask pinpoints the corresponding pixels of the fish object). Our proposed CNN model is subdivided into three parts: (1) the base trunk, (2) the region-proposal network, (3) the pixel segmentation network, and (4) the objectness detector and box-coordinate predictor network.

2.1 Base Trunk Subnetwork

In most CNN architectures, the base trunk consists of a very deep network responsible for generating data-driven features to be fed to object detector and segmentation networks. As per [18], a deep network can generate meaningful features starting from basic RGB values through several convolution operators that generate features from neighboring spatial information. Unlike traditional neural networks, each layer in a CNN has three dimensions (height, width and depth), corresponding to a feature-map. Experimental results show that the first layers of deep networks generate patterns on edges of small regions of the input frame. Later layers combine several small patterns to form larger and more meaningful shapes (i.e. fish eyes, fins, mouth).

For this model, we design the base trunk according to the inception-resnet-v2 architecture [22] that consists of more than a hundred layers. It has achieved state-of-the-art results in the ImageNet challenge. Beginning with the input layer, the base-trunk passes the input frame through several inception blocks. As the layers progress, the height, and width of the feature map decrease, while the depth increases. Following [18], features generated at the very end of the base-trunk are general enough to be useful for a wide variety of tasks, and can be combined for both localization and segmentation. To save memory and for computational efficiency, the base trunk shrinks the original input image to size $H/16 \times W/16 \times D$, where $D = 512$ as shown in Fig. 1.

Similar to [18], the base trunk network does not have any sub-loss function on its own. Weights in the base trunk merely receive gradient corrections from the proposal, segmentation and box coordinate predictor sub-networks.

2.2 Region Proposal Subnetwork

The region proposal network (RPN) is responsible for generating hundreds of region proposals per input frame. A region proposal is a box-coordinate set $[x_1, y_1, x_2, y_2]$ corresponding to a sub-region in the input frame, that is likely to contain a fish-object. Some region proposal methods rely on external superpixel

proposal algorithms [6], [20]. But the Faster R-CNN work of [18], incorporates the proposal stage as a subnetwork task in the overall CNN architecture. The RPN in Faster R-CNN relies on “anchors”, which are spaced evenly every 16 pixels. Each anchor location is assigned with 9 boxes of different sizes and aspect ratios. The RPN operates by finding those anchor-boxes that have the highest intersection over union (or IoU) over ground-truth bounding boxes, as well as those that have an IoU above 70%. Each anchor-box outputs a regressed box coordinate set $[x1, y1, x2, y2]$ that roughly approximates the actual coordinates of the object it overlaps. But these coordinates are usually not accurate, and [18] resorts to a secondary coordinate refinement procedure through the R-CNN sub-network that is connected with the RPN.

RPNs are trained using a loss function formed from a linear combination of objectness classification losses and smooth L1 loss functions [6]. Formally, let $F(\Theta)$ denote the shareable feature map received from the base trunk subnetwork of size of $H/16 \times W/16$, and where Θ denotes all network parameters. Similar to the Faster R-CNN RPN architecture, the RPN passes a 3×3 convolution over $F(\Theta)$, followed by two 1×1 convolutions to produce $p_i^a(\Theta)$ and $t_i^a(\Theta)$, where i denotes an anchor location. The RPN’s compound loss function L_i^{rpn} for a certain anchor location i is as follows:

$$L_i^{rpn} = l_{cls}(p_i^a(\Theta)) + l_{reg}(t_i^a(\Theta)) \quad (1)$$

where $p_i^a(\Theta)$ is an 18-d vector of object probabilities for anchor i ’s 9 scale and aspect ratios. $t_i^{anchor}(\Theta)$ is a 36-d vector of bounding box parameter adjustments. l_{cls} is softmax, while l_{reg} is the smoothL1 loss function. Both $p_i^a(\Theta)$ and $t_i^{anchor}(\Theta)$ involve two inputs relating to ground-truth inputs and network prediction outputs. The details over $t_i^{anchor}(\Theta)$ are a bit involved but are all written in [6] and [18]. The loss function in Eq. 1 is summed up over all i anchor boxes, which reaches an average of 12,000 boxes, followed by averaging. We let A define the set of all anchor box locations such that for each anchor $i \in A$. The RPN loss function is then:

$$L^{rpn} = \frac{1}{|A|} \sum_i L_i^{rpn} \quad (2)$$

Bounding box parameter adjustments $t_i^a(\Theta)$ can be transformed to bounding box coordinates $[x1, y1, x2, y2]$ using a box-inversion function shown in [6]. For each anchor i , the corresponding regressed bounding box coordinates $[x1, y1, x2, y2]$ are used by the R-CNN in [18] to extract sub-feature maps from $F(\Theta)$. This procedure is described in more detail in the following subsections.

2.3 Segmentation Subnetwork

The segmentation task consists of labeling each pixel in the input frame as either belonging to a fish-object or not. The expected output is a binary mask where pixels of fish objects are correspondingly labeled as 1 (or foreground), and all other pixels are labeled as 0 (or background). For this model, segmentation is

first performed in the entire frame through a fully-convolutional network (or FCN) [16]. Segmentation could be performed on a local per-region basis as has been done in [3] or [7]. But we prefer to perform global segmentation through an FCN subnetwork since it is able to benefit from a combination of both global and local information. From [16] FCN takes features generated at the very last feature map of the base trunk (with $H/16 \times W/16$ dimensions) and estimates a foreground-background per-pixel probability map with the original $H \times W$ input size. Estimation of the foreground-background probability map is performed through a series of bilinear-interpolation and mini inception blocks. Bilinear-interpolation is responsible for restoring the small $H/16 \times W/16$ base trunk end-feature map to the original $H \times W$ input size.

The segmentation sub-network has its own set of loss functions. We let $F^{fcn}(\Theta)$ describe the FCN output tensor of size $H \times W \times 2$ that is normalized to form a probability defined over foreground and background classes (2 channels). To make the notation simple, we let $|F^{fcn}(\Theta)|$ denote the number of pixels corresponding to $F^{fcn}(\Theta)$. The loss function is then:

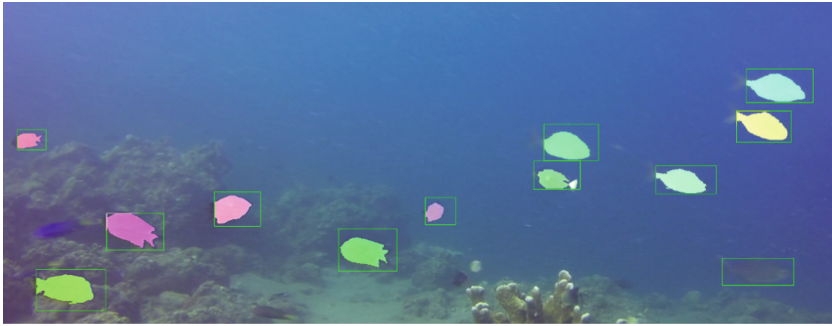


Fig. 2. Large frame fish detection and instance segmentation

$$L^{seg} = \frac{1}{|F^{fcn}(\Theta)|} \sum_{|F^{fcn}(\Theta)|} \log(F^{fcn}(\Theta))I[fg] \tag{3}$$

From Eq. 3, L^{seg} is the cross entropy of the $F^{fcn}(\Theta)$ foreground/background probability predictions per pixel with the true label $I[fg]$ which takes the value of 1 if foreground and 0 o.w. The resulting foreground-background per-pixel probability map provides the probability that a certain pixel in the input frame belongs to a foreground fish object (or not). However, the FCN output map does not yet differentiate whether a certain pixel belongs to a certain fish object or to another. In other words, the predictions are instance-insensitive. To assign foreground pixels among the various fish objects detected is the role of the objectness detector and box-coordinate predictor network.

2.4 Dense Conditional Random Field Post-processing and Fish Object Instance Differentiation

Before combining the mask and box-coordinate outputs from the segmentation and box-coordinate subnetworks respectively, we first perform an additional mask refinement scheme. Specifically, after the FCN subnetwork produces a 2-channel foreground/background probability mask for all pixels in the input frame, the mask is passed through a dense conditional random field (CRF) for post-processing [9]. This has been performed as a mask segmenter refinement procedure in [2]. The dense CRF associates the foreground/background per-pixel probabilities to the RGB color-space of the input frame to come up with more refined contours and shapes of fish objects. Unary potentials in the dense CRF are the log outputs of the FCN segmentation sub-network $\log(F^{fcn}(\theta))$. Pairwise potentials consist of euclidian distances between pixels in the RGB space following the sample from [9]. By using RGB values, learning dense CRF pairwise potentials is greatly simplified, and inference can be done linearly using convolutional filters following from [9] in a transformed permutahedral space. After dense CRF post-processing, each pixel is assigned to its corresponding fish object using the box coordinates produced by the box coordinate predictor sub-network (BCPS). This creates an instance-sensitive classification of the pixel mask.

2.5 Objectness Detector and Box Coordinate Predictor Subnetwork

The box coordinate predictor subnetwork (BCPS) follows the refinement process in Faster R-CNN [18], but with additional cascades that perform stepwise corrections on box predictions. The BCPS subnetwork begins with extraction of sub-feature maps from the shareable feature map $F(\theta)$. For each predicted anchor location $p_i^a(\theta)$ with $\geq 70\%$ objectness probability, the BCPS subnetwork uses RoI-Cropping [3] to extract a set of sub feature maps $f(\theta)$ from $F(\theta)$. The initial dimensions of $f(\theta)$ are dependent on the regressed anchor box coordinates $[x1, y1, x2, y2]$ from the RPN, but RoI-cropping bilinearly interpolates all $f(\theta)$ to a uniform $h \times w$ - following the R-CNN structure in [18].

We model the BCPS subnetwork to follow a 4-cascade process [3] for better accuracy. The first cascade predicts preliminary box-coordinates of fish objects along with their objectness information, where high objectness scores represent a region-of-interest with a high probability of containing a fish species. The next three cascades serve to refine the box-coordinates predicted in the previous step. For increased accuracy, each subnetwork incorporates local mask information as well following the architecture described in [3]. In mathematical notation, we let $t_c(f_c(\theta))$ denote the bounding box parameter adjustments for cascade c . Using the inversion function shown in [6], these bounding box adjustments are transformed to bounding box coordinates $[x1, y1, x2, y2]$. Using these coordinates, the next cascade $c + 1$ re-extracts sub-feature maps $f_{c+1}(\theta)$ and passes them on to the next cascade sub-network for another regression that outputs the

next bounding box adjustments $t_{c+1}(f_{c+1}(\Theta))$. Following the cascade algorithm in [3], bounding box adjustments in $t_{c+1}(f_{c+1}(\Theta))$ should contain a tighter box around the fish object.

For the loss function of BCPS, we let f denote the set of sub-feature maps extracted by the BCPS sub-network where $f_c(\Theta) \in f$ for all cascades $c = 1 \dots 4$. The sub loss function of the BCPS sub-network is a summation of losses contributed by each subfeature map, with a total of $4|f|$ sub-feature maps (given 4 cascades). Similar to the RPN, the losses are composed of a smooth L1 regression loss $l_{reg}^c(t_c(f_c(\Theta)))$ and a softmax cross-entropy loss $l_{cls}^c(p_c(f_c(\Theta)))$. The sub-loss function of the BCPS sub-network is as follows:

$$L^{bcps} = \frac{1}{4|f|} \sum_{c=1}^4 \sum_f \left[l_{reg}^c(t_c(f_c(\Theta))) + l_{cls}^c(p_c(f_c(\Theta))) \right] \quad (4)$$

Where c defines the position of the cascade, which in this case begins from 1 to 4. As per [3], incorporating more cascades in the box coordinate regression process should provide more accuracy gains.

2.6 Total Loss Function

During training of the network, the overall loss function supervises gradient corrections for all network parameters Θ , using losses formed in a linear combination of the sub-loss functions (Eq. 5). We train the network using stochastic gradient descent (SGD) with learning rate 0.001 and Nesterov Momentum with 0.9 momentum parameter.

$$L^{all} = L^{rpn} + L^{seg} + L^{bcps} \quad (5)$$

2.7 Prediction Outputs

After training, the network outputs a set of results during prediction (test) time as listed below. The 1st and 2nd outputs are from the BCPS subnetwork (as the RPN subnetwork outputs are merely intermediate outputs for generating proposals). The 3rd output uses the FCN global mask and takes only those pixels that are identified to be within a box coordinate set $[x1, y1, x2, y2]$.

1. Coordinates $[x1, y1, x2, y2]$ of pixels with high probabilities of containing fish objects ($\geq 70\%$)
2. Probability of fish objectness given each coordinate set $[x1, y1, x2, y2]$
3. Per pixel foreground-background probability for each coordinate set $[x1, y1, x2, y2]$, following FCN and dense CRF post-processing.

2.8 Data Sources and Training Details

The network is trained with benthic underwater video frames taken from ten (10) different sites in central Philippines. Each video frame has different background,

underwater illumination, and hues. Some video frames have static camera setups while some video frames experience rapid camera movement. In some sites, only a few fish species can be visually detected, but other sites have more than 20 fish objects per frame. To train the proposed model, the needed information are the box coordinates of fish objects per frame (to train the object and coordinate predictor subnetwork) along with the corresponding ground truth binary mask per frame (to train the fully convolutional segmentation sub-network).

The usual method to train a deep learning model is through manually-labeled ground-truth. However, given the large number of frames that have to be processed, it is impractical and time-consuming to manufacture manually-labeled ground truth. We settled for training the model with weakly-labeled data derived from motion-based background subtraction tools. We first extract motion masks from an adaptive Gaussian background subtractor implemented in OpenCV [1], and then refine it using a 152-layer fully convolutional network [15]. Afterwards, we pass the refined motion masks to a contour detector to extract fish blobs. The detected fish blobs are then passed to a deep network classifier for prediction of the true fish blobs.

We select 1,600 weakly-labeled training frames for the training pool, where each site has at least 10 frames included. Training of the network is done with a 12 GB Titan X GPU, with tensorflow deep learning computer code. Training time took around four (4) days, since the network went over the entire training set 80 times.

3 Results and Discussion

Figure 2 depicts a sample frame taken from one of the sites, where fish objects are simultaneously enclosed in bounding boxes and segmented pixel-wise. For testing the model, we collected 30 frames randomly sampled from the different sites and calculated their precision and recall values in terms of bounding box-localization. We consider a detected fish object as a true positive (TP) if the bounding box visually covers more than 50% of the fish's pixel area. If two bounding boxes cover the same fish, only one is counted as a true positive, while the duplicate box is penalized as a false positive. These rules are similar to the standard PASCAL VOC localization challenge [4]. We manually as many fish objects as we could find in the 30 test-frames, but some fish objects that are very small and blurry are no longer counted as they already lie very far from the camera center (Table 1).

Upon test-time, we gathered a precision of 93.77% and a recall of 48.51%, where the average number of fish objects detected per frame number up to 12 fishes per frame. We would like to note however, that the 48.51% recall measure includes fishes of all sizes. However, this measure includes fish objects that are already located far from the camera, and have pixel sizes that are already $\leq 30 \times 30$. If the experiment's scope is limited to fishes that are within a certain perimeter radius from the camera, i.e. a 5-meter box perimeter radius, then the recall of fish objects should be much higher, and we have a conservative estimate



Fig. 3. Sample fish detections with simultaneous bounding box and pixel instance segmentation. The figures above depict sample detections produced by the model. The model simultaneously estimates both the fish object's bounding box (green box outline) and segments the pixels belonging to the fish object within the bounding box. The results above include Dense CRF post-processing which facilitate more accurate segmentation. Segmentation is robust whether the fish object is located in the water column or along the seabed with coral background (i.e. the bottom left picture) (Color figure online)

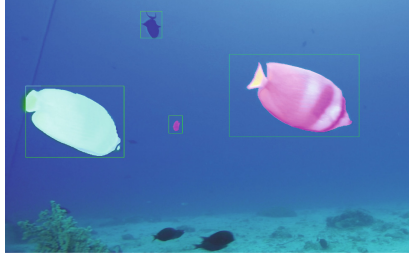


Fig. 4. Sample detection with moving camera. Since our model predicts on a frame-by-frame basis, it does not need temporal information to localize and segment fish objects

Table 1. Set A includes fishes of all sizes discernable by a human observer. Several of the fishes in set A however are very small with pixel sizes $\leq 30 \times 30$. Set B only includes fishes that have pixel sizes that are roughly $\geq 100 \times 100$ and are seen within a 5-meter volume radius from the camera viewpoint. Given this constraint, recall increases to 60% estimate. We consider an object to be correctly detected if the predicted bounding box is 50% the area of the ground-truth box, following PASCAL VOC.

Test set	Precision	Recall
Set A	93.77	48.51
Set B	93.77	60 (est.)

of more than 60% recall. This recall includes most of the larger fishes of pixel sizes $\geq 100 \times 100$.

From our experiments on test-set frames, several of the detected fish objects lie along the middle portion of the frame’s y-axis. This section is where the contrast between the fish objects and the water column are easier to detect. Several of the missed fish objects lie along the frame’s base, where most of the corals are located. Several of these missed fish objects are hard to detect even from a human observer. Most of the large and noticeable fishes however are adequately detected and segmented by the model as shown in Fig. 3. The results here include dense CRF post-processing, which greatly increases segmentation precision. If we compare our model with segmentation results pre CRF post-processing, the segmentation with CRF post-processing has more noticeable edges and very finely outlines the fish’s contour relative to either a water column or seabed background. Our model is generally high in precision and 93.77% of the bounding boxes are able to cover the fish object by more than 50% - analogously matching the PASCAL VOC requirement for a true-positive in terms of box area coverage.

We tested the model in another set of test frames with rapid camera movement, and in a completely different site. Our model is able to localize and segment the fishes. A sample is in Fig. 4, where individual fishes were segmented. Recall and precision are lower however than the original test set since the frames are from completely different environments. But it is notable that the model is

able to operate despite camera movements which shows its independence from motion-based information. Independence from camera movements, shows that our network model can have applications to videos taken from unmanned water vehicles, where jittery camera movements are expected.

4 Conclusion

In this paper, we proposed a deep network model that is able to simultaneously localize fish objects, estimate their bounding boxes and segment their corresponding pixels using a unified multi-task convolutional neural network. Estimation is independent of motion-based information and processing can be performed on a frame-by-frame basis. The model can detect a wide variety of fish object shapes and sizes against different backgrounds. Our proposed model provides a solution for the problem of fish counting in marine environments through its localization feature, which allows automated counting of fish organisms represented by their bounding boxes. Precision is high at 93.77%, with a 60% estimated recall for fish within a 5-meter volume radius. In addition, the added segmentation feature of our model provides a visual description of fish object contours - which would be useful in aiding the visual verification of a human expert or for future work related to fish size estimation.

References

1. Bradski, G., Kaehler, A.: *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media Inc., Sebastopol (2008)
2. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. arXiv preprint [arXiv:1606.00915](https://arxiv.org/abs/1606.00915) (2016)
3. Dai, J., He, K., Sun, J.: Instance-aware semantic segmentation via multi-task network cascades. arXiv preprint [arXiv:1512.04412](https://arxiv.org/abs/1512.04412) (2015)
4. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The Pascal visual object classes (VOC) challenge. *Int. J. Comput. Vis.* **88**(2), 303–338 (2010)
5. Fier, R., Albu, A.B., Hoeberechts, M.: Automatic fish counting system for noisy deep-sea videos. In: *Oceans-St. John's 2014*, pp. 1–6. IEEE (2014)
6. Girshick, R.: Fast R-CNN. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448 (2015)
7. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988. IEEE (2017)
8. Joly, A., et al.: Lifeclef: multimedia life species identification. In: *EMR@ ICMR*, pp. 7–13 (2014)
9. Koltun, V.: Efficient inference in fully connected CRFs with Gaussian edge potentials. *Adv. Neural Inf. Process. Syst.* **2**(3), 4 (2011)
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*, pp. 1097–1105 (2012)
11. Kumar Rai, R., Gour, P., Singh, B.: Underwater image segmentation using clahe enhancement and thresholding. *Int. J. Emerg. Technol. Adv. Eng.* **2**(1), 118–123 (2012)

12. Labao, A.B., Naval, P.C.: Weakly-labelled semantic segmentation of fish objects in underwater videos using a deep residual network. In: Nguyen, N.T., Tojo, S., Nguyen, L.M., Trawiński, B. (eds.) ACIIDS 2017. LNCS (LNAI), vol. 10192, pp. 255–265. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-54430-4_25
13. Li, X., Shang, M., Hao, J., Yang, Z.: Accelerating fish detection and recognition by sharing CNNs with objectness learning. In: OCEANS 2016-Shanghai, pp. 1–5. IEEE (2016)
14. Li, X., Shang, M., Qin, H., Chen, L.: Fast accurate fish detection and recognition of underwater images with fast R-CNN. In: OCEANS 2015-MTS/IEEE Washington, pp. 1–5. IEEE (2015)
15. Lin, G., Shen, C., van den Hengel, A., Reid, I.: Efficient piecewise training of deep structured models for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3194–3203 (2016)
16. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)
17. Negahdaripour, S., Yu, C.H.: On shape and range recovery from image shading for underwater applications. *Underwater Robot. Veh.: Des. Control* 221–250 (1995)
18. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99 (2015)
19. Russakovsky, O., et al.: Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **115**(3), 211–252 (2015)
20. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint [arXiv:1312.6229](https://arxiv.org/abs/1312.6229) (2013)
21. Spampinato, C., Chen-Burger, Y.H., Nadarajan, G., Fisher, R.B.: Detecting, tracking and counting fish in low quality unconstrained underwater videos. *VISAPP* **2**(2008), 514–519 (2008)
22. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.: Inception-v4, inception-resnet and the impact of residual connections on learning. arXiv preprint [arXiv:1602.07261](https://arxiv.org/abs/1602.07261) (2016)
23. Twilley, N., Graber, C.: Gastropod: How many fish are in the sea? counting fish is a daunting but essential task in protecting aquatic ecosystems-and now artificial intelligence, autonomous submarines, and drones can help. <https://www.theatlantic.com/science/archive/2016/10/how-many-fish-are-in-the-sea/502937/>